Software Engineering Institute Publishes *Software Technology Review*

A Cliffs Notes Approach for PEOs, PMs, IPTs, and Support Staff

ROBERT ROSENSTEIN • KIMBERLY BRUNE • JOHN FOREMAN

Carnegie Mellon University
Software Engineering Institute
Software Technology Review

The Software Technology Review takes information that is in the minds of experts and makes it available to everyone.

As a program executive officer, do you sit in meetings and wonder about the new technology being discussed? Are you comfortable the Request for Proposal (RFP) prepared by your staff will be clearly understood by potential respondents? Would you want your program manager (PM) to risk recommending a new technology for your organization without fully knowing its limitations and alternatives?

The *Software Technology Review*, through a ground-breaking project undertaken by the Software Engineering Institute, answers these questions. Now existing as a hard copy document as well as an active World Wide Web site, the *Software Technology Review* is a reference source that catalogs existing and emerging software technologies.

Motivation for Development

Work on the *Software Technology Review* was initiated in early 1996 when Darleen Druyun, Principal Deputy Assistant Secretary of the Air Force for Acquisition, asked the Software Engineering Institute to produce a prototype Software Technology Reference Guide that would provide information for the Air Force to plan research, development, and technology transition to satisfy DoD mission needs.

Primary Resource

Since meeting those objectives with the initial publication of a hard copy document in January 1997, the *Software Technology Review* has broadened its scope. In an effort to provide a primary source of information about software technology, we documented a shared common-knowledge base and provided a collection of high-level information that points to in-depth information.

To the best of our knowledge, a collection of this kind of information does not exist. In numerous cases, people are not aware of many of these technologies. Even if they are familiar with them, their perceptions are often off the mark. Currently, if you need information about a specific technology, you would ask the experts for their opinion; the *Software Technology Review* takes information that is in the minds of experts and makes it available to everyone. Our work minimizes the need to search extensively for this kind of information and, in turn, makes it easier to make educated decisions about software technology.

Rosenstein is the Software Technology Review *Project Coordinator, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa. Brune is the Technical Writer-Editor for the* Software Technology Review *and Foreman is the Managing Editor. Both are members of the* Software Technology Review *Publications Staff, Software Engineering Institute.*

Benefits

A wide diversity of government-industry managers and executives use and promote the *Software Technology Review*:

- Executives find use of the *Software Technology Review* enables them to prepare better presentations and speeches because they have a more complete understanding of software technologies.

- Technical investigators use the *Software Technology Review* to point to documented experiences of use.

- Systems programmers utilize the *Software Technology Review* to obtain information that will enable them to properly evaluate proposals.

- Contractors benefit from the *Software Technology Review* by using the technology descriptions as a guide/reference baseline in their proposal writing.

- Organizations consult the *Software Technology Review* to capture a broad picture of the state of the practice.

Goals

The *Software Technology Review* is intended to be a reference source to specific software technologies of interest. The document has many goals, including —

- encapsulating a large amount of information so that the Program Executive Officer (PEO) or PM can rapidly read the basics and make a preliminary decision on whether further investigation is warranted;

- achieving objectivity, balance, and a quantitative focus, bringing out shortcomings as well as advantages;

- providing insight into areas such as cost, risk, quality, ease of use, security, and alternatives; and

- pointing to references and sources of more detailed information, including usage and experience.

Limitations

While the *Software Technology Review* strives to provide balanced coverage of a wide scope of technology, certain constraints restrict the content in the following areas:

- Not prescriptive. The *Software Technology Review* does not make recommendations, establish priorities, or dictate a specific path or approach.

- Not a product reference. The *Software Technology Review* is not a survey or catalog of products.

- Not an endorsement. Inclusion or exclusion of a topic in the *Software Technology Review* does not constitute an endorsement of any type, or selection as any sort of "best technical practice."

- Not a market forecasting tool. While the technology description may project the effects of a technology and discuss trends, other organizations produce more complete analysis and forecast reports.

- Not a focused analysis of specific technical areas. Various sources offer reports on a subscription or one-time basis and may also produce specialized analyses and reporting on a consulting basis.

Target Audiences

We developed the *Software Technology Review* to be used by PEOs, PMs, Integrated Product Teams (IPT), and their support staff in the following manner:

- Technology Transfer And Technology Insertion Guidelines

- Overview/Introductory Information

- Baseline Reference Document

- *Cliffs Notes* Approach (Provides High-Level, Four- to Six-Page Quick Study)

- Trade-off Information

- Taxonomies to Aid in Identifying Alternatives

- Back Pointers to High-Level, Related Technologies

- Criteria and Guidance for Decision Making

Current Availability

Prior to publication of the *Software Technology Review*, the first "official" release of this reference document was the *C4 Software Technology Reference Guide – A Prototype*, first published by the Software Engineering Institute and industry participants for the U.S. Air Force acquisition community in January 1997. Since then, our World Wide Web site became operational. The site has the most current technology descriptions as well as the latest Portable Document Format (PDF) and Postscript version of the document.

We Want Your Participation

The *Software Technology Review* is modeled after professional refereed journals (i.e., *Communications of the ACM* [Association for Computing Machinery], *IEEE Software* [Institute of Electrical and Electronics Engineers]), with volunteer authors, reviewers, or editorial board members. The Software Engineering Institute provides the overall management and coordination of the *Software Technology Review.*

The *Software Technology Review* team invites you to volunteer as a credited author, reviewer, maintainer, or editorial board member. With your participation, we can enhance the *Software Technology Review's* relevance and generate widespread community interest in its long-term development and maintenance.

Don't Hesitate to Contact Us

For more information, we invite you to visit our World Wide Web site at **http://www.sei.cmu.edu.** You may also contact Robert Rosenstein, project coordinator, at (412) 268-8468, or by E-mail at **str@sei.cmu.edu**.

TECHNOLOGY DESCRIPTION OVERVIEW

The purpose of a technology description is to identify a technology, characterize it in terms of the property of systems and measures of software quality that it affects, and point out trade-offs, benefits, risks, and limitations that may arise in various scenarios of use.

Each technology description also provides reference(s) to literature, indications of current maturity of the technology, and cross references to related technologies. Technology descriptions are not meant to be comprehensive. Each technology description provides the PM with enough knowledge to decide whether to investigate further, to find out where to go for more information, and to know what questions to ask in gathering more information.

Status. An assessment of the overall quality and maturity of the technology description.

Note. Prerequisite readings that provide an overview of the general topic area and establish a context for the different technologies in the area.

Purpose and Origin. General description and brief background of the technology. Includes what capability or benefit was anticipated when originally conceived, cites quality measures that are significantly influenced by the technology, and identifies common aliases as well as its originators or key developers.

Technical Detail. Answers the question, “What does the technology do?” Includes the salient quality measures that are influenced by the technology in all situations and describes trade-offs that are enabled.

Usage Considerations. Example applications into which this technology may or may not be incorporated and quality measures that may be influenced by this technology.

Maturity. An indication as to how well the technology is developed.

Costs and Limitations. Limitations and costs of using a particular technology; includes investments in other technologies, time, or money. Indicates a direct conflict with security or real-time requirements.

Dependencies. Other technologies that significantly influence or are significantly influenced by the technology.

Alternatives. An alternative technology is one that could be used for the same purposes as the technology being described.

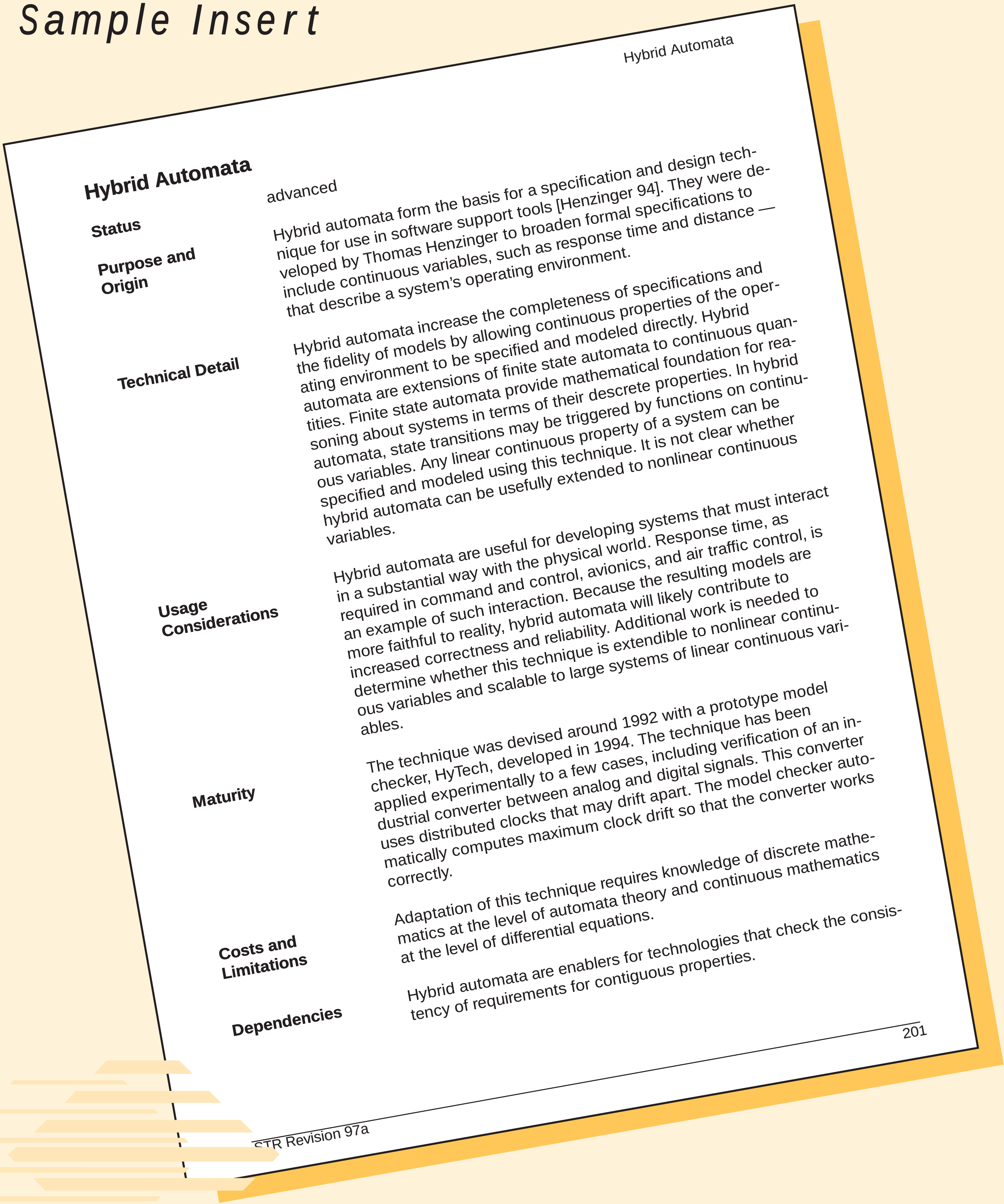
Complementary Technologies. A complementary technology is one that enhances or is enhanced by the technology being described, but for which neither is critical to the development or use of the other.

Index Categories. Keywords under which this technology is indexed:

- Application category. How this technology would be employed, either in support of operational systems or in actual operations of systems.
- Quality Measures category. Quality attributes (e.g., reliability or responsiveness) that are influenced in some way by the application of this technology.
- Computing Reviews category. Technical sub-discipline within computer science into which the technology falls.

SOFTWARE TECH

Technical Description — Sample Insert



Hybrid Automata

Status

advanced

Purpose and Origin

Hybrid automata form the basis for a specification and design technique for use in software support tools [Henzinger 94]. They were developed by Thomas Henzinger to broaden formal specifications to include continuous variables, such as response time and distance — that describe a system's operating environment.

Technical Detail

Hybrid automata increase the completeness of specifications and the fidelity of models by allowing continuous properties of the operating environment to be specified and modeled directly. Hybrid automata are extensions of finite state automata to continuous quantities. Finite state automata provide mathematical foundation for reasoning about systems in terms of their descrete properties. In hybrid automata, state transitions may be triggered by functions on continuous variables. Any linear continuous property of a system can be specified and modeled using this technique. It is not clear whether hybrid automata can be usefully extended to nonlinear continuous variables.

Usage Considerations

Hybrid automata are useful for developing systems that must interact in a substantial way with the physical world. Response time, as required in command and control, avionics, and air traffic control, is an example of such interaction. Because the resulting models are more faithful to reality, hybrid automata will likely contribute to increased correctness and reliability. Additional work is needed to determine whether this technique is extendible to nonlinear continuous variables and scalable to large systems of linear continuous variables.

Maturity

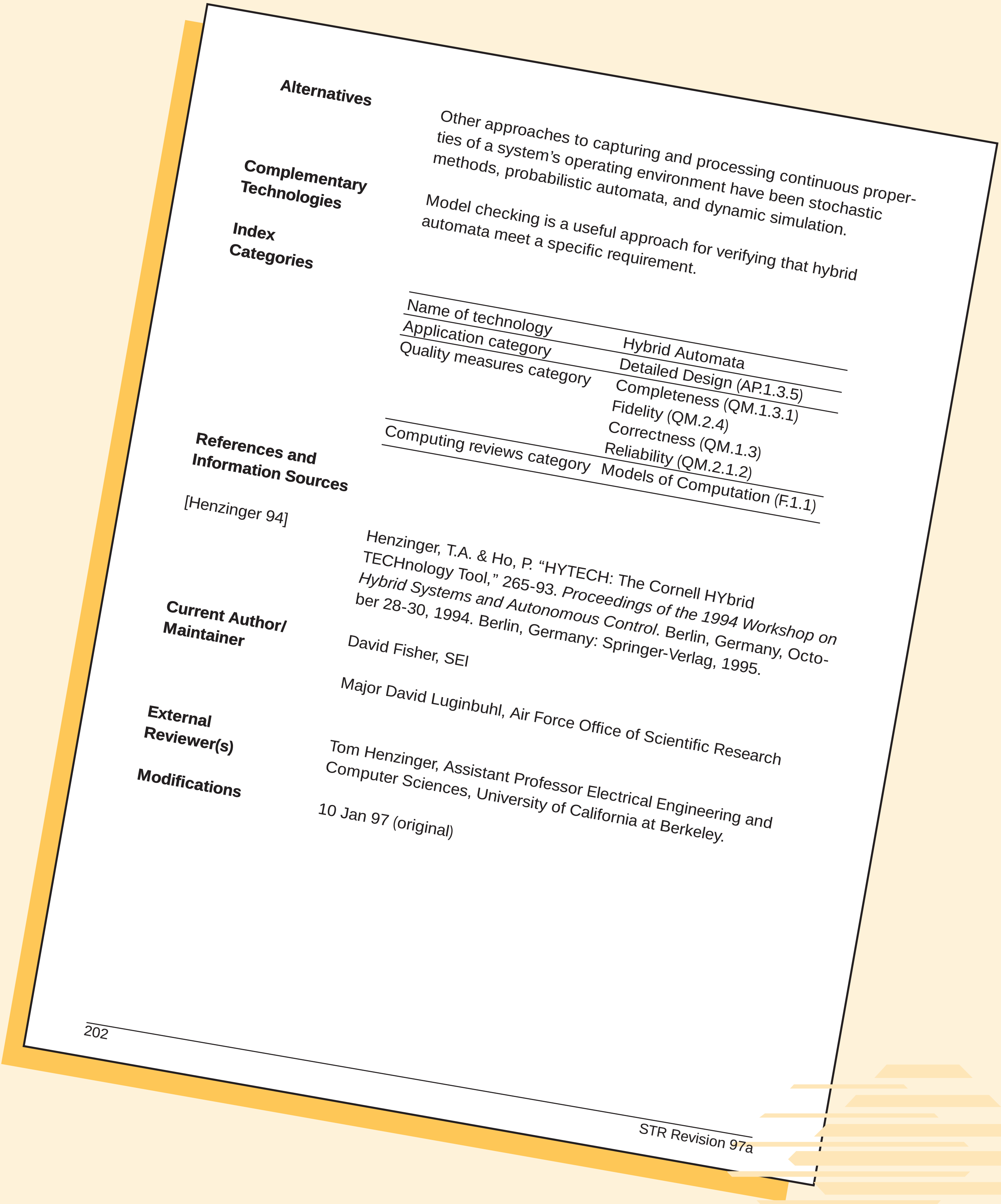
The technique was devised around 1992 with a prototype model checker, HyTech, developed in 1994. The technique has been applied experimentally to a few cases, including verification of an industrial converter between analog and digital signals. This converter uses distributed clocks that may drift apart. The model checker automatically computes maximum clock drift so that the converter works correctly.

Costs and Limitations

Adaptation of this technique requires knowledge of discrete mathematics at the level of automata theory and continuous mathematics at the level of differential equations.

Dependencies

Hybrid automata are enablers for technologies that check the consistency of requirements for contiguous properties.



Alternatives

Other approaches to capturing and processing continuous properties of a system's operating environment have been stochastic methods, probabilistic automata, and dynamic simulation.

Complementary Technologies

Model checking is a useful approach for verifying that hybrid automata meet a specific requirement.

Index Categories

Name of technology	Hybrid Automata
Application category	Detailed Design (AP.1.3.5)
Quality measures category	Completeness (QM.1.3.1) Fidelity (QM.2.4) Correctness (QM.1.3) Reliability (QM.2.1.2)
Computing reviews category	Models of Computation (F.1.1)

References and Information Sources

[Henzinger 94] Henzinger, T.A. & Ho, P. "HYTECH: The Cornell HYbrid TECHnology Tool," 265-93. *Proceedings of the 1994 Workshop on Hybrid Systems and Autonomous Control*. Berlin, Germany, October 28-30, 1994. Berlin, Germany: Springer-Verlag, 1995.

Current Author/ Maintainer

David Fisher, SEI

External Reviewer(s)

Major David Luginbuhl, Air Force Office of Scientific Research

Tom Henzinger, Assistant Professor Electrical Engineering and Computer Sciences, University of California at Berkeley.

Modifications

10 Jan 97 (original)

202

STR Revision 97a